PASSNYC MAIN

Manoranjan Kumar

7/23/2018

# Read the saved scv back again  
SchoolMain <- read.csv("https://raw.githubusercontent.com/manusatx/NYC\_SCHOOLS/master/SchoolMain.csv")  
dim(SchoolMain)  
## [1] 1272 52

# data exploration for predictors Remove ID and name and other Identificable  
# varaibles by looking at the data  
SchoolMain <- subset(SchoolMain, select = -c(X, District, New., Adjusted.Grade,   
 GEOID, Other.Location.Code.in.LCGMS, SchoolGEOID, School.Name, SED.Code,   
 Location.Code, Latitude, Longitude, Address..Full., City, Zip, Grades, Grade.Low,   
 Grade.High, Rigorous.Instruction.Rating, Collaborative.Teachers.Rating,   
 Supportive.Environment.Rating, Effective.School.Leadership.Rating, Strong.Family.Community.Ties.Rating,   
 Trust.Rating, Student.Achievement.Rating))  
dim(SchoolMain)  
## [1] 1272 27

# Convert community to 1 / 0 vs Yes No  
SchoolMain$Community.School. <- ifelse(SchoolMain$Community.School. == "Yes",   
 1, 0)  
  
# Convert all columns to numeric  
SchoolMain <- as.data.frame(lapply(SchoolMain, function(x) as.numeric(as.character(x))))  
## Warning in FUN(X[[i]], ...): NAs introduced by coercion  
  
## Warning in FUN(X[[i]], ...): NAs introduced by coercion  
  
## Warning in FUN(X[[i]], ...): NAs introduced by coercion  
  
## Warning in FUN(X[[i]], ...): NAs introduced by coercion  
  
## Warning in FUN(X[[i]], ...): NAs introduced by coercion  
  
## Warning in FUN(X[[i]], ...): NAs introduced by coercion  
  
## Warning in FUN(X[[i]], ...): NAs introduced by coercion  
  
## Warning in FUN(X[[i]], ...): NAs introduced by coercion  
  
## Warning in FUN(X[[i]], ...): NAs introduced by coercion  
  
## Warning in FUN(X[[i]], ...): NAs introduced by coercion  
  
## Warning in FUN(X[[i]], ...): NAs introduced by coercion  
  
## Warning in FUN(X[[i]], ...): NAs introduced by coercion

# Make Community.School as factor  
SchoolMain$Community.School. <- as.factor(SchoolMain$Community.School.)  
  
# Merge the ELA and maths score to form one response variable column  
SchoolMain$Average\_Proficiency <- (SchoolMain$Average.ELA.Proficiency + SchoolMain$Average.Math.Proficiency)/2  
SchoolMain <- subset(SchoolMain, select = -c(Average.ELA.Proficiency, Average.Math.Proficiency))  
  
# Missing values  
summary(SchoolMain)  
## Community.School. Economic.Need.Index School.Income.Estimate  
## 0:1196 Min. :0.0500 Min. : 16902   
## 1: 76 1st Qu.:0.5500 1st Qu.: 33610   
## Median :0.7300 Median : 43151   
## Mean :0.6728 Mean : 48443   
## 3rd Qu.:0.8400 3rd Qu.: 58518   
## Max. :0.9600 Max. :181382   
## NA's :25 NA's :396   
## Percent.ELL Percent.Asian Percent.Black Percent.Hispanic  
## Min. :0.0000 Min. :0.0000 Min. :0.0000 Min. :0.0200   
## 1st Qu.:0.0400 1st Qu.:0.0100 1st Qu.:0.0600 1st Qu.:0.1800   
## Median :0.0900 Median :0.0400 Median :0.2400 Median :0.3550   
## Mean :0.1248 Mean :0.1165 Mean :0.3200 Mean :0.4115   
## 3rd Qu.:0.1700 3rd Qu.:0.1400 3rd Qu.:0.5525 3rd Qu.:0.6400   
## Max. :0.9900 Max. :0.9500 Max. :0.9700 Max. :1.0000   
##   
## Percent.Black...Hispanic Percent.White Student.Attendance.Rate  
## Min. :0.0300 Min. :0.0000 Min. :0.0000   
## 1st Qu.:0.4900 1st Qu.:0.0100 1st Qu.:0.9200   
## Median :0.9000 Median :0.0300 Median :0.9400   
## Mean :0.7314 Mean :0.1316 Mean :0.9272   
## 3rd Qu.:0.9600 3rd Qu.:0.1600 3rd Qu.:0.9500   
## Max. :1.0000 Max. :0.9200 Max. :1.0000   
## NA's :25   
## Percent.of.Students.Chronically.Absent Rigorous.Instruction..  
## Min. :0.0000 Min. :0.0000   
## 1st Qu.:0.1100 1st Qu.:0.8600   
## Median :0.2000 Median :0.9000   
## Mean :0.2157 Mean :0.8948   
## 3rd Qu.:0.3000 3rd Qu.:0.9400   
## Max. :1.0000 Max. :1.0000   
## NA's :25 NA's :25   
## Collaborative.Teachers.. Supportive.Environment..  
## Min. :0.0000 Min. :0.0000   
## 1st Qu.:0.8500 1st Qu.:0.8400   
## Median :0.9000 Median :0.8900   
## Mean :0.8844 Mean :0.8875   
## 3rd Qu.:0.9400 3rd Qu.:0.9400   
## Max. :1.0000 Max. :1.0000   
## NA's :25 NA's :25   
## Effective.School.Leadership.. Strong.Family.Community.Ties..  
## Min. :0.0000 Min. :0.0000   
## 1st Qu.:0.7600 1st Qu.:0.8000   
## Median :0.8300 Median :0.8300   
## Mean :0.8162 Mean :0.8309   
## 3rd Qu.:0.8900 3rd Qu.:0.8700   
## Max. :0.9900 Max. :0.9900   
## NA's :25 NA's :25   
## Trust.. NumOfLibraries NumOfAfterSchoolProgs  
## Min. :0.0000 Min. : 0.000 Min. : 0.00   
## 1st Qu.:0.8700 1st Qu.: 2.000 1st Qu.:13.00   
## Median :0.9200 Median : 3.000 Median :25.00   
## Mean :0.9042 Mean : 3.169 Mean :29.77   
## 3rd Qu.:0.9400 3rd Qu.: 4.000 3rd Qu.:47.00   
## Max. :1.0000 Max. :10.000 Max. :80.00   
## NA's :25   
## NumOfLowIncomeUnits MedianHldIncome IncomeToPovertyRatio  
## Min. : 0.00 Min. : 12052 Min. : 0   
## 1st Qu.: 1.00 1st Qu.: 31492 1st Qu.: 658   
## Median : 15.00 Median : 47518 Median : 930   
## Mean : 34.41 Mean : 53655 Mean :1037   
## 3rd Qu.: 62.00 3rd Qu.: 68854 3rd Qu.:1314   
## Max. :141.00 Max. :250001 Max. :7004   
## NA's :8   
## ChildPoverty CountofSNAPHlds HealthCoverage Average\_Proficiency  
## Min. : 0.0 Min. : 0.0 Min. : 0 Min. :1.895   
## 1st Qu.: 263.0 1st Qu.: 593.2 1st Qu.: 1553 1st Qu.:2.275   
## Median : 751.0 Median : 1339.0 Median : 3667 Median :2.515   
## Mean : 830.6 Mean : 1473.0 Mean : 3814 Mean :2.602   
## 3rd Qu.:1243.0 3rd Qu.: 2000.0 3rd Qu.: 5595 3rd Qu.:2.860   
## Max. :5420.0 Max. :12683.0 Max. :28186 Max. :4.040   
## NA's :55

# check only for response variable Yield  
SchoolMain[is.na(SchoolMain$Average\_Proficiency), ]  
## Community.School. Economic.Need.Index School.Income.Estimate  
## 8 0 NA 28552.64  
## 13 0 0.65 41943.27  
## 16 0 0.85 25982.42  
## 22 0 0.86 26385.67  
## 24 0 NA NA  
## 37 0 NA 25720.46  
## 52 0 0.90 28756.98  
## 106 0 0.88 24333.94  
## 121 0 NA 26999.15  
## 128 0 0.87 22255.01  
## 145 0 0.93 20772.06  
## 150 0 0.83 28339.16  
## 164 0 NA 46890.41  
## 167 0 0.93 25044.32  
## 168 0 0.92 25806.36  
## 202 0 0.86 27037.25  
## 321 0 0.43 70496.26  
## 336 0 NA NA  
## 376 0 NA 47993.87  
## 416 0 0.71 38458.24  
## 446 0 NA 42721.54  
## 448 0 NA 39389.02  
## 464 0 0.86 34016.72  
## 467 0 NA 40623.06  
## 476 0 0.90 37260.80  
## 536 0 NA 41476.71  
## 541 0 NA NA  
## 549 0 0.67 44501.44  
## 553 0 0.80 40077.78  
## 645 0 NA 51874.13  
## 653 0 NA 53389.56  
## 674 0 NA 35778.71  
## 682 1 0.94 NA  
## 709 0 0.73 34019.94  
## 719 0 NA 118322.46  
## 748 0 NA 86857.50  
## 769 0 NA 70655.41  
## 780 0 NA NA  
## 782 0 NA 87424.06  
## 812 0 NA 30958.57  
## 855 0 0.54 42375.71  
## 856 0 0.79 34892.30  
## 863 0 NA 50890.96  
## 874 0 0.74 36111.12  
## 910 0 0.62 44717.99  
## 938 0 NA NA  
## 1026 0 0.57 71741.52  
## 1040 0 NA 42821.68  
## 1053 0 NA NA  
## 1066 0 0.62 46867.48  
## 1077 0 0.74 50347.75  
## 1117 0 0.13 69328.45  
## 1123 0 0.58 41791.27  
## 1137 0 0.52 48735.89  
## 1200 0 0.31 63770.09  
## Percent.ELL Percent.Asian Percent.Black Percent.Hispanic  
## 8 0.06 0.00 0.45 0.52  
## 13 0.02 0.05 0.48 0.43  
## 16 0.25 0.00 0.29 0.69  
## 22 0.12 0.01 0.27 0.72  
## 24 0.14 0.00 0.34 0.63  
## 37 0.21 0.00 0.26 0.71  
## 52 0.23 0.00 0.33 0.62  
## 106 0.16 0.00 0.40 0.57  
## 121 0.05 0.01 0.51 0.45  
## 128 0.15 0.01 0.31 0.68  
## 145 0.17 0.00 0.39 0.59  
## 150 0.22 0.00 0.22 0.78  
## 164 0.25 0.16 0.16 0.64  
## 167 0.54 0.01 0.10 0.87  
## 168 0.28 0.01 0.22 0.73  
## 202 0.08 0.01 0.43 0.51  
## 321 0.03 0.05 0.33 0.30  
## 336 0.04 0.05 0.22 0.24  
## 376 0.06 0.01 0.67 0.24  
## 416 0.03 0.01 0.70 0.25  
## 446 0.02 0.02 0.89 0.06  
## 448 0.01 0.01 0.91 0.07  
## 464 0.07 0.01 0.74 0.21  
## 467 0.02 0.00 0.90 0.06  
## 476 0.05 0.02 0.74 0.22  
## 536 0.14 0.02 0.25 0.63  
## 541 0.08 0.14 0.14 0.33  
## 549 0.06 0.03 0.32 0.54  
## 553 0.21 0.02 0.31 0.65  
## 645 0.01 0.00 0.94 0.06  
## 653 0.00 0.00 0.91 0.06  
## 674 0.03 0.01 0.75 0.23  
## 682 0.07 0.00 0.63 0.34  
## 709 0.09 0.06 0.26 0.63  
## 719 0.05 0.16 0.03 0.13  
## 748 0.02 0.19 0.20 0.21  
## 769 0.07 0.14 0.16 0.52  
## 780 0.02 0.04 0.10 0.18  
## 782 0.01 0.15 0.02 0.19  
## 812 0.11 0.04 0.28 0.63  
## 855 0.02 0.06 0.39 0.28  
## 856 0.15 0.01 0.28 0.69  
## 863 0.09 0.01 0.32 0.30  
## 874 0.04 0.01 0.55 0.40  
## 910 0.16 0.02 0.11 0.64  
## 938 0.17 0.02 0.21 0.73  
## 1026 0.00 0.02 0.92 0.03  
## 1040 0.35 0.05 0.01 0.92  
## 1053 0.19 0.23 0.38 0.32  
## 1066 0.58 0.62 0.00 0.35  
## 1077 0.23 0.07 0.02 0.78  
## 1117 0.01 0.37 0.04 0.20  
## 1123 0.56 0.86 0.02 0.08  
## 1137 0.28 0.74 0.03 0.15  
## 1200 0.13 0.53 0.03 0.18  
## Percent.Black...Hispanic Percent.White Student.Attendance.Rate  
## 8 0.97 0.02 NA  
## 13 0.91 0.03 0.95  
## 16 0.98 0.01 0.92  
## 22 0.99 0.00 0.91  
## 24 0.97 0.01 NA  
## 37 0.97 0.01 NA  
## 52 0.95 0.03 0.92  
## 106 0.98 0.02 0.91  
## 121 0.96 0.01 NA  
## 128 0.98 0.00 0.94  
## 145 0.98 0.01 0.88  
## 150 0.99 0.01 0.96  
## 164 0.80 0.03 NA  
## 167 0.97 0.01 0.91  
## 168 0.95 0.03 0.92  
## 202 0.94 0.02 0.95  
## 321 0.63 0.29 1.00  
## 336 0.46 0.47 NA  
## 376 0.91 0.06 NA  
## 416 0.95 0.01 1.00  
## 446 0.95 0.02 NA  
## 448 0.98 0.01 NA  
## 464 0.95 0.02 0.91  
## 467 0.96 0.01 NA  
## 476 0.96 0.02 0.92  
## 536 0.88 0.09 NA  
## 541 0.47 0.36 NA  
## 549 0.85 0.09 1.00  
## 553 0.96 0.01 0.95  
## 645 1.00 0.00 NA  
## 653 0.97 0.00 NA  
## 674 0.97 0.00 NA  
## 682 0.97 0.02 0.89  
## 709 0.89 0.02 0.92  
## 719 0.16 0.59 NA  
## 748 0.41 0.35 NA  
## 769 0.68 0.16 NA  
## 780 0.27 0.63 NA  
## 782 0.21 0.59 NA  
## 812 0.92 0.03 NA  
## 855 0.67 0.20 0.95  
## 856 0.97 0.01 0.94  
## 863 0.62 0.34 NA  
## 874 0.95 0.03 0.96  
## 910 0.75 0.21 0.95  
## 938 0.94 0.04 NA  
## 1026 0.95 0.01 0.95  
## 1040 0.94 0.01 NA  
## 1053 0.70 0.02 NA  
## 1066 0.35 0.02 0.96  
## 1077 0.81 0.12 0.94  
## 1117 0.23 0.37 0.97  
## 1123 0.10 0.03 0.97  
## 1137 0.18 0.05 0.96  
## 1200 0.21 0.23 0.95  
## Percent.of.Students.Chronically.Absent Rigorous.Instruction..  
## 8 NA NA  
## 13 0.10 0.99  
## 16 0.25 0.91  
## 22 0.33 0.97  
## 24 NA NA  
## 37 NA NA  
## 52 0.32 1.00  
## 106 0.36 0.70  
## 121 NA NA  
## 128 0.17 0.95  
## 145 0.53 0.92  
## 150 0.09 0.96  
## 164 NA NA  
## 167 0.33 0.98  
## 168 0.30 0.96  
## 202 0.14 0.83  
## 321 0.00 0.90  
## 336 NA NA  
## 376 NA NA  
## 416 0.00 0.91  
## 446 NA NA  
## 448 NA NA  
## 464 0.34 0.91  
## 467 NA NA  
## 476 0.28 0.99  
## 536 NA NA  
## 541 NA NA  
## 549 0.00 0.97  
## 553 0.14 0.94  
## 645 NA NA  
## 653 NA NA  
## 674 NA NA  
## 682 0.41 0.98  
## 709 0.31 0.93  
## 719 NA NA  
## 748 NA NA  
## 769 NA NA  
## 780 NA NA  
## 782 NA NA  
## 812 NA NA  
## 855 0.12 0.98  
## 856 0.72 0.89  
## 863 NA NA  
## 874 0.05 0.97  
## 910 0.10 0.95  
## 938 NA NA  
## 1026 0.10 0.88  
## 1040 NA NA  
## 1053 NA NA  
## 1066 0.07 0.90  
## 1077 0.19 0.92  
## 1117 0.02 0.93  
## 1123 0.04 0.88  
## 1137 0.12 0.93  
## 1200 0.14 0.99  
## Collaborative.Teachers.. Supportive.Environment..  
## 8 NA NA  
## 13 0.98 1.00  
## 16 0.95 1.00  
## 22 0.96 1.00  
## 24 NA NA  
## 37 NA NA  
## 52 1.00 0.99  
## 106 0.76 0.90  
## 121 NA NA  
## 128 0.89 0.95  
## 145 0.95 0.96  
## 150 0.97 0.99  
## 164 NA NA  
## 167 0.97 0.98  
## 168 0.95 0.96  
## 202 0.86 0.96  
## 321 0.92 1.00  
## 336 NA NA  
## 376 NA NA  
## 416 0.93 0.99  
## 446 NA NA  
## 448 NA NA  
## 464 0.97 0.96  
## 467 NA NA  
## 476 1.00 0.99  
## 536 NA NA  
## 541 NA NA  
## 549 0.96 1.00  
## 553 0.98 1.00  
## 645 NA NA  
## 653 NA NA  
## 674 NA NA  
## 682 0.96 0.97  
## 709 0.89 0.97  
## 719 NA NA  
## 748 NA NA  
## 769 NA NA  
## 780 NA NA  
## 782 NA NA  
## 812 NA NA  
## 855 0.99 0.99  
## 856 0.79 0.93  
## 863 NA NA  
## 874 0.96 0.99  
## 910 0.99 0.96  
## 938 NA NA  
## 1026 0.88 0.95  
## 1040 NA NA  
## 1053 NA NA  
## 1066 0.81 0.92  
## 1077 0.94 0.95  
## 1117 0.85 0.96  
## 1123 0.96 0.99  
## 1137 0.92 0.99  
## 1200 0.97 0.99  
## Effective.School.Leadership.. Strong.Family.Community.Ties.. Trust..  
## 8 NA NA NA  
## 13 0.93 0.84 0.94  
## 16 0.93 0.91 0.99  
## 22 0.84 0.89 0.98  
## 24 NA NA NA  
## 37 NA NA NA  
## 52 0.98 0.91 0.99  
## 106 0.67 0.82 0.84  
## 121 NA NA NA  
## 128 0.78 0.78 0.95  
## 145 0.95 0.87 0.96  
## 150 0.95 0.91 0.99  
## 164 NA NA NA  
## 167 0.93 0.86 0.97  
## 168 0.94 0.87 0.94  
## 202 0.84 0.82 0.91  
## 321 0.84 0.95 0.96  
## 336 NA NA NA  
## 376 NA NA NA  
## 416 0.85 0.95 0.95  
## 446 NA NA NA  
## 448 NA NA NA  
## 464 0.96 0.92 0.99  
## 467 NA NA NA  
## 476 0.96 0.92 0.99  
## 536 NA NA NA  
## 541 NA NA NA  
## 549 0.86 0.97 0.99  
## 553 0.92 0.90 0.98  
## 645 NA NA NA  
## 653 NA NA NA  
## 674 NA NA NA  
## 682 0.97 0.90 0.98  
## 709 0.87 0.83 0.94  
## 719 NA NA NA  
## 748 NA NA NA  
## 769 NA NA NA  
## 780 NA NA NA  
## 782 NA NA NA  
## 812 NA NA NA  
## 855 0.97 0.94 0.99  
## 856 0.73 0.88 0.86  
## 863 NA NA NA  
## 874 0.94 0.95 0.99  
## 910 0.97 0.94 0.97  
## 938 NA NA NA  
## 1026 0.77 0.86 0.91  
## 1040 NA NA NA  
## 1053 NA NA NA  
## 1066 0.77 0.77 0.90  
## 1077 0.91 0.89 0.97  
## 1117 0.78 0.88 0.88  
## 1123 0.94 0.87 0.98  
## 1137 0.70 0.85 0.86  
## 1200 0.91 0.89 0.98  
## NumOfLibraries NumOfAfterSchoolProgs NumOfLowIncomeUnits  
## 8 5 61 67  
## 13 3 36 62  
## 16 2 52 40  
## 22 4 59 59  
## 24 2 53 42  
## 37 3 51 49  
## 52 4 71 91  
## 106 3 48 64  
## 121 4 65 122  
## 128 5 56 128  
## 145 4 67 62  
## 150 5 58 111  
## 164 4 14 3  
## 167 6 61 119  
## 168 6 61 119  
## 202 3 12 1  
## 321 3 24 30  
## 336 4 21 36  
## 376 4 40 14  
## 416 4 48 53  
## 446 5 35 54  
## 448 3 40 35  
## 464 4 47 83  
## 467 2 40 34  
## 476 2 51 94  
## 536 2 26 13  
## 541 3 12 3  
## 549 2 33 7  
## 553 4 55 32  
## 645 2 18 12  
## 653 2 19 3  
## 674 2 26 90  
## 682 3 54 97  
## 709 4 66 31  
## 719 4 37 6  
## 748 9 38 22  
## 769 6 15 6  
## 780 4 15 6  
## 782 4 32 25  
## 812 5 63 100  
## 855 7 52 93  
## 856 7 66 91  
## 863 9 56 105  
## 874 8 60 121  
## 910 3 48 28  
## 938 3 17 9  
## 1026 1 3 0  
## 1040 4 19 3  
## 1053 2 15 4  
## 1066 2 22 0  
## 1077 2 36 7  
## 1117 1 7 0  
## 1123 2 15 1  
## 1137 2 15 1  
## 1200 2 3 0  
## MedianHldIncome IncomeToPovertyRatio ChildPoverty CountofSNAPHlds  
## 8 16503 1196 1396 1960  
## 13 49180 1353 922 2391  
## 16 26066 944 1247 1347  
## 22 28537 1261 1699 1913  
## 24 28537 1261 1699 1913  
## 37 21551 1278 1790 2044  
## 52 42713 847 388 2175  
## 106 33661 880 1055 1302  
## 121 17500 1179 1191 2000  
## 128 19912 765 1000 1118  
## 145 30281 903 1353 1308  
## 150 31396 941 1017 1561  
## 164 40221 800 716 1150  
## 167 23947 1278 1312 1834  
## 168 23947 1278 1312 1834  
## 202 74861 249 262 436  
## 321 128989 660 375 1624  
## 336 75682 395 236 720  
## 376 45909 339 288 676  
## 416 16727 758 592 1183  
## 446 52019 595 528 1411  
## 448 48375 776 1116 1354  
## 464 39572 976 764 1967  
## 467 47239 1240 1541 2000  
## 476 40612 1264 1712 1938  
## 536 71042 363 180 853  
## 541 82935 1181 941 1914  
## 549 42325 990 719 2281  
## 553 31493 835 1991 968  
## 645 72813 450 407 644  
## 653 60417 591 642 720  
## 674 37917 732 1312 1025  
## 682 28164 888 1386 1429  
## 709 21102 601 342 978  
## 719 85426 1310 453 3693  
## 748 126036 835 422 2755  
## 769 71772 886 152 3933  
## 780 52823 1535 1063 4215  
## 782 82026 3340 2297 7857  
## 812 20288 900 1208 2022  
## 855 36759 942 920 2144  
## 856 51538 517 352 1892  
## 863 48429 1258 1243 2876  
## 874 38276 498 650 1308  
## 910 30882 492 475 734  
## 938 27675 2088 2346 3488  
## 1026 69408 847 30 242  
## 1040 46979 775 27 154  
## 1053 41322 882 73 165  
## 1066 60026 1017 11 65  
## 1077 55952 893 0 84  
## 1117 76550 1222 0 48  
## 1123 36663 1610 50 242  
## 1137 46038 2956 25 365  
## 1200 83322 768 0 100  
## HealthCoverage Average\_Proficiency  
## 8 4794 NA  
## 13 5344 NA  
## 16 3906 NA  
## 22 6405 NA  
## 24 6405 NA  
## 37 6011 NA  
## 52 3878 NA  
## 106 4350 NA  
## 121 5301 NA  
## 128 3580 NA  
## 145 4267 NA  
## 150 4429 NA  
## 164 3302 NA  
## 167 5127 NA  
## 168 5127 NA  
## 202 1190 NA  
## 321 3411 NA  
## 336 1650 NA  
## 376 1559 NA  
## 416 2749 NA  
## 446 3446 NA  
## 448 3817 NA  
## 464 4889 NA  
## 467 5643 NA  
## 476 5474 NA  
## 536 1766 NA  
## 541 4504 NA  
## 549 5538 NA  
## 553 4202 NA  
## 645 1863 NA  
## 653 2748 NA  
## 674 3692 NA  
## 682 4176 NA  
## 709 2670 NA  
## 719 6230 NA  
## 748 5113 NA  
## 769 6681 NA  
## 780 7682 NA  
## 782 15502 NA  
## 812 4383 NA  
## 855 5788 NA  
## 856 3332 NA  
## 863 7036 NA  
## 874 3036 NA  
## 910 2730 NA  
## 938 9304 NA  
## 1026 176 NA  
## 1040 438 NA  
## 1053 628 NA  
## 1066 117 NA  
## 1077 294 NA  
## 1117 38 NA  
## 1123 337 NA  
## 1137 933 NA  
## 1200 47 NA

print("There are 55 rows with some missing data")  
## [1] "There are 55 rows with some missing data"

# Remove 55 rows from data where response variable is NULL  
SchoolMain\_NA\_Resp <- SchoolMain[is.na(SchoolMain$Average\_Proficiency), ]  
SchoolMain <- SchoolMain[!is.na(SchoolMain$Average\_Proficiency), ]  
  
# see predictors data with missing values  
nrow(SchoolMain[rowSums(is.na(subset(SchoolMain, select = -Average\_Proficiency))) >   
 0, ])  
## [1] 394

print("There are 394 rows with some missing data")  
## [1] "There are 394 rows with some missing data"

#Use Knn imputation to preprocess the data  
library(caret)  
SchoolMain\_prepro\_index <- preProcess(subset(SchoolMain, select = - Average\_Proficiency),   
 #method=c("BoxCox", "center", "scale", "knnImpute"))  
 method=c("BoxCox", "knnImpute"))  
SchoolMain\_new <- predict(SchoolMain\_prepro\_index, SchoolMain)  
  
#see predictors data with missing values after imputation  
nrow(SchoolMain\_new[rowSums(is.na(subset(SchoolMain\_new, select = - Average\_Proficiency))) > 0,])  
## [1] 0

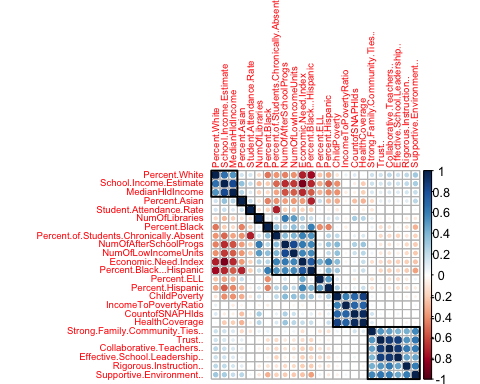
print('There are 0 rows with some missing data after knn imputation')  
## [1] "There are 0 rows with some missing data after knn imputation"

#Create a df with predictors only and without the factor columns has 24 columns  
SchoolMain\_new\_numeric <- SchoolMain\_new[,sapply(SchoolMain\_new[, !names(SchoolMain\_new) %in% c("Average\_Proficiency")], is.numeric)]  
  
#df with factor and response has 2 columns   
SchoolMain\_fctr\_resp <- subset(SchoolMain\_new, select=c(Community.School.,Average\_Proficiency))

# check for near zero variance variables  
library(caret)  
rm\_SchoolMain\_cols <- nearZeroVar(SchoolMain\_new)  
rm\_SchoolMain\_cols  
## integer(0)

print("there is no columns with near zero variance")  
## [1] "there is no columns with near zero variance"

# Check the data numeric volumns for correlation for 24 predictors  
library(caret)  
SchoolMaincorr = cor(SchoolMain\_new\_numeric)  
corrplot::corrplot(SchoolMaincorr, order = "hclust", addrect = 9, method = "circle",   
 tl.cex = 0.6)

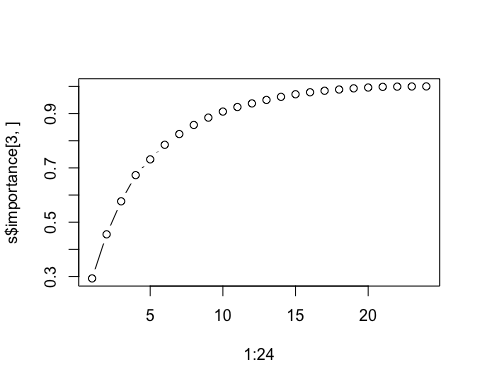


# library(ggcorrplot) ggcorrplot(SchoolMaincorr ,hc.order = TRUE, method =  
# c('square'),tl.cex = 7, show.diag = FALSE, ggtheme =  
# ggplot2::theme\_minimal)  
  
# Check for correlated varaibles and remove highly correlated predictors  
SchoolMaincorrpp <- findCorrelation(SchoolMaincorr, cutoff = 0.8)  
SchoolMain\_no\_cor <- SchoolMain\_new\_numeric[, -SchoolMaincorrpp]  
dim(SchoolMain\_new\_numeric)  
## [1] 1217 24

dim(SchoolMain\_no\_cor)  
## [1] 1217 17

print("From the continuous variables 7 correlated predictor variables were removed with correlation check and now we have 24-7 = 17 predictor variables")  
## [1] "From the continuous variables 7 correlated predictor variables were removed with correlation check and now we have 24-7 = 17 predictor variables"

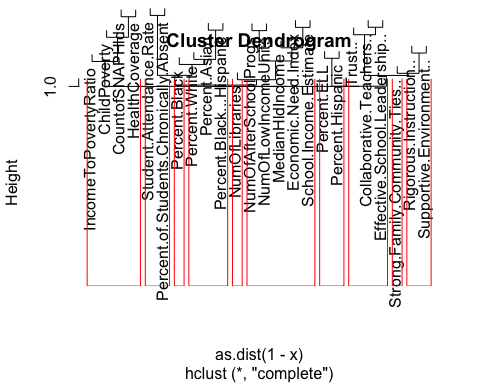
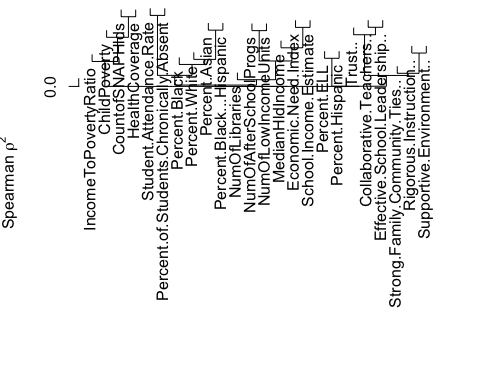
# PCA  
SchoolMain.pca <- prcomp(SchoolMain\_new\_numeric, center = TRUE, scale. = TRUE)  
s <- summary(SchoolMain.pca)  
# Plot cumulative variance Principal components  
plot(1:24, s$importance[3, ], type = "b")



print("PC1, PC2 till PC10 explain 90% of variance cumulatively as seen from the summary")  
## [1] "PC1, PC2 till PC10 explain 90% of variance cumulatively as seen from the summary"

# Plot the resultant Principal components biplot(SchoolMain.pca, scale = 1,  
# cex = 0.6)  
  
# with removed highly correlated variables datset is now having 19 columns  
# and 1217 rows  
SchoolMain\_reducedcorr <- cbind(SchoolMain\_no\_cor, SchoolMain\_fctr\_resp)  
dim(SchoolMain\_reducedcorr)  
## [1] 1217 19

# variable clustering and dimesnionality reduction calculate R2 Ratio for  
# each variable in each cluster. We will then select variable with minimum  
# 1-R2 ratio in each cluster as cluster representative  
r2clusterfun <- function(inputdf, insim, numcuttree) {  
 library(Hmisc)  
 varclus.inputdf <- varclus(data.matrix(inputdf), similarity = insim)  
 plot(varclus.inputdf)  
 # Below is hclust object  
 hclust.inputdf <- varclus.inputdf$hclust  
 # plot the 10 clusters by cut  
 plot(hclust.inputdf)  
 rect.hclust(hclust.inputdf, k = numcuttree, border = "red")  
 # Cut the above into 10 clusters and assign name to data frame  
 groups.inputdf <- as.data.frame(cutree(hclust.inputdf, k = numcuttree))  
 library(data.table)  
 groups.inputdf <- as.data.frame(setDT(groups.inputdf, keep.rownames = TRUE)[])  
 colnames(groups.inputdf) <- c("var", "clusternum")  
 groups.inputdf$ID <- seq.int(nrow(groups.inputdf))  
 groups.inputdf$ID <- as.character(groups.inputdf$ID)  
 groups.inputdf.index.list <- as.list(as.matrix(by(groups.inputdf$ID, groups.inputdf$clusternum,   
 function(x) return(as.numeric(as.character(x))))))  
 # Check for index lists > 1  
 index.list <- groups.inputdf.index.list[sapply(groups.inputdf.index.list,   
 length) > 1]  
 index.list.1 <- groups.inputdf.index.list[sapply(groups.inputdf.index.list,   
 length) == 1]  
 # Disimilarity matrix (1-r^2) using Pearson  
 cor.inputdf <- cor(inputdf, method = insim)  
 cormatrix <- round((1 - (cor.inputdf)^2), 3)  
 # cormatrix is dissimilarity matrix Dissimalrity (1-r^2) between elements of  
 # a cluster and the other elements in its own cluster  
 h <- function(index) {  
 temp <- cormatrix[index, index]  
 diag(temp) <- NA  
 apply(temp, 1, min, na.rm = T)  
 }  
 numer <- lapply(index.list, h)  
 # Dissimalrity (1-r^2) between elements of each cluster and other clusters  
 g <- function(index) {  
 apply(cormatrix[-index, index], 2, min)  
 }  
 denom <- lapply(index.list, g)  
 # Find the minimum r^2 ratio  
 i <- function(index) {  
 which.min(numer[[index]]/denom[[index]])  
 }  
 apply(as.matrix(1:length(index.list)), 1, i)  
 # get the index of each cluster lowest r2 element  
 min\_r2\_index <- as.data.frame(apply(as.matrix(1:length(index.list)), 1,   
 i))  
 colnames(min\_r2\_index) <- c("min\_r2\_idx")  
 min\_r2\_index$id <- seq.int(nrow(min\_r2\_index))  
   
 print("The 10 variables with the lowest r2 ratio in each cluster using lowest rsq ratio is :")  
 subsetvar <- matrix(NA, nrow = length(index.list), ncol = 1)  
 for (i in min\_r2\_index[, "id"]) {  
 j <- min\_r2\_index[i, "min\_r2\_idx"]  
 i\_idx <- index.list[[i]][j]  
 subsetvar[i] <- as.character(groups.inputdf[groups.inputdf$ID == i\_idx,   
 ][, "var"])  
 }  
 # get columns for clusters which had only one variable  
 index.list.1.df <- as.data.frame(unlist(index.list.1))  
 colnames(index.list.1.df) <- c("col\_idx")  
 index.list.1.df$id <- seq.int(nrow(index.list.1.df))  
 subsetvarone <- matrix(NA, nrow = length(index.list.1), ncol = 1)  
 for (i in index.list.1.df[, "id"]) {  
 i\_idx <- index.list.1.df[i, "col\_idx"]  
 subsetvarone[i] <- as.character(groups.inputdf[groups.inputdf$ID ==   
 i\_idx, ][, "var"])  
 }  
 allsubvars <- rbind(subsetvar, subsetvarone)  
 return(allsubvars)  
}  
# Apply function r2clusterfun use cut = 10 as seen from pca around PC1 to  
# PC10 explian 90% of variance  
SchoolMain\_subset <- r2clusterfun(SchoolMain\_new\_numeric, insim = "spearman",   
 numcuttree = 10)



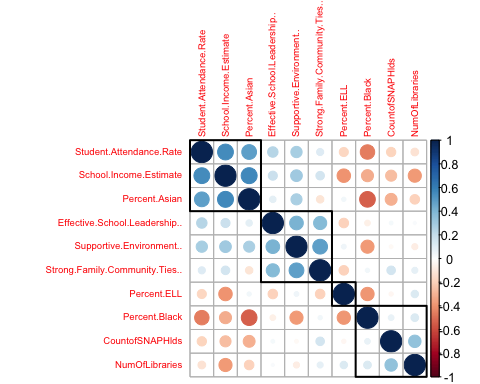
## [1] "The 10 variables with the lowest r2 ratio in each cluster using lowest rsq ratio is :"

# These are the predictor variables selected after numcuttree = 10 and min  
# rsq ratio selection.  
print(SchoolMain\_subset[, 1])  
## [1] "School.Income.Estimate" "Percent.ELL"   
## [3] "Percent.Asian" "Student.Attendance.Rate"   
## [5] "Supportive.Environment.." "Effective.School.Leadership.."   
## [7] "CountofSNAPHlds" "Percent.Black"   
## [9] "Strong.Family.Community.Ties.." "NumOfLibraries"

print("correlation between the above 10 elements is")  
## [1] "correlation between the above 10 elements is"

# Reduced dataset with only 10 variables is SchoolMain\_reduced  
SchoolMain\_reduced\_numeric <- SchoolMain\_new\_numeric[, c(SchoolMain\_subset)]  
SchoolMain\_subset\_corr <- round(cor(SchoolMain\_reduced\_numeric, method = c("spearman")),   
 2)  
# SchoolMain\_subset\_corr  
print("The subset of variables selected show no or very low correlation to each other.")  
## [1] "The subset of variables selected show no or very low correlation to each other."

# Check the data numeric volumns for correlation  
corrplot::corrplot(SchoolMain\_subset\_corr, order = "hclust", addrect = 4, method = "circle",   
 tl.cex = 0.6)



# Create a data set with reduced predictor variables with  
SchoolMain\_reduced <- cbind(SchoolMain\_reduced\_numeric, SchoolMain\_fctr\_resp)  
dim(SchoolMain\_reduced)  
## [1] 1217 12

# split the unreduced data into train and test  
set.seed(1000)  
train.school.index <- createDataPartition(y = SchoolMain\_reducedcorr$Average\_Proficiency,   
 p = 0.85, list = FALSE)  
trainschool <- SchoolMain\_reducedcorr[train.school.index, ]  
testschool <- SchoolMain\_reducedcorr[-train.school.index, ]  
dim(trainschool)  
## [1] 1036 19

dim(testschool)  
## [1] 181 19

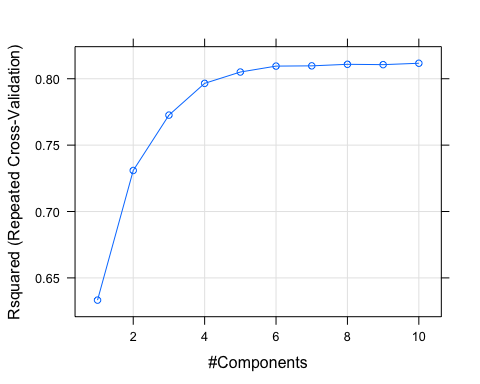
# split the reduced data into train and test  
set.seed(1000)  
train.reduced.school.index <- createDataPartition(y = SchoolMain\_reduced$Average\_Proficiency,   
 p = 0.85, list = FALSE)  
trainreducedschool <- SchoolMain\_reduced[train.reduced.school.index, ]  
testreducedschool <- SchoolMain\_reduced[-train.reduced.school.index, ]  
dim(trainreducedschool)  
## [1] 1036 12

dim(testreducedschool)  
## [1] 181 12

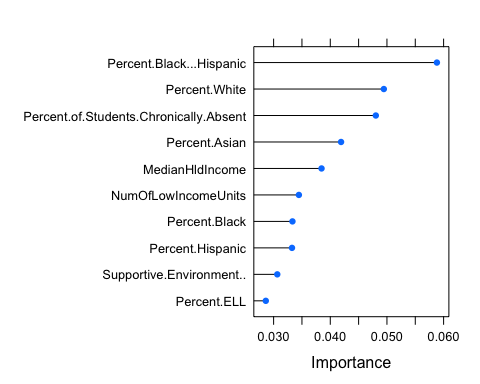
# Run the LM model  
set.seed(1000)  
lm\_school\_model = train(Average\_Proficiency ~ ., data = trainschool, method = "lm",   
 tuneLength = 10, preProcess = c("center", "scale"), trControl = trainControl(method = "repeatedcv",   
 number = 10))  
summary(lm\_school\_model)  
##   
## Call:  
## lm(formula = .outcome ~ ., data = dat)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.57977 -0.11099 -0.00548 0.09219 1.00472   
##   
## Coefficients:  
## Estimate Std. Error t value  
## (Intercept) 2.602268 0.005474 475.404  
## Percent.ELL -0.121449 0.007711 -15.749  
## Percent.Asian 0.107265 0.031750 3.378  
## Percent.Black -0.168512 0.032467 -5.190  
## Percent.Hispanic -0.108706 0.027549 -3.946  
## Percent.Black...Hispanic 0.067095 0.038719 1.733  
## Percent.White 0.077912 0.035330 2.205  
## Student.Attendance.Rate -0.185993 0.008546 -21.765  
## Percent.of.Students.Chronically.Absent -0.233970 0.010540 -22.198  
## Rigorous.Instruction.. 0.031818 0.007815 4.071  
## Supportive.Environment.. 0.034309 0.008925 3.844  
## Effective.School.Leadership.. -0.017209 0.006940 -2.480  
## Strong.Family.Community.Ties.. 0.013931 0.007706 1.808  
## NumOfLibraries 0.021097 0.006711 3.143  
## NumOfLowIncomeUnits 0.020326 0.008521 2.385  
## MedianHldIncome -0.009525 0.007974 -1.195  
## IncomeToPovertyRatio -0.009542 0.008182 -1.166  
## HealthCoverage 0.012484 0.009011 1.385  
## Community.School.1 -0.022932 0.005870 -3.907  
## Pr(>|t|)   
## (Intercept) < 2e-16 \*\*\*  
## Percent.ELL < 2e-16 \*\*\*  
## Percent.Asian 0.000757 \*\*\*  
## Percent.Black 2.53e-07 \*\*\*  
## Percent.Hispanic 8.49e-05 \*\*\*  
## Percent.Black...Hispanic 0.083419 .   
## Percent.White 0.027660 \*   
## Student.Attendance.Rate < 2e-16 \*\*\*  
## Percent.of.Students.Chronically.Absent < 2e-16 \*\*\*  
## Rigorous.Instruction.. 5.04e-05 \*\*\*  
## Supportive.Environment.. 0.000128 \*\*\*  
## Effective.School.Leadership.. 0.013305 \*   
## Strong.Family.Community.Ties.. 0.070928 .   
## NumOfLibraries 0.001718 \*\*   
## NumOfLowIncomeUnits 0.017245 \*   
## MedianHldIncome 0.232547   
## IncomeToPovertyRatio 0.243801   
## HealthCoverage 0.166228   
## Community.School.1 9.98e-05 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.1762 on 1017 degrees of freedom  
## Multiple R-squared: 0.8191, Adjusted R-squared: 0.8159   
## F-statistic: 255.9 on 18 and 1017 DF, p-value: < 2.2e-16

# Run the LM model with pca  
set.seed(1000)  
lm\_pca\_school\_model = train(Average\_Proficiency ~ ., data = trainschool, method = "lm",   
 tuneLength = 10, preProcess = c("pca"), trControl = trainControl(method = "repeatedcv",   
 number = 10))  
summary(lm\_pca\_school\_model)  
##   
## Call:  
## lm(formula = .outcome ~ ., data = dat)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.50479 -0.14165 -0.02542 0.10982 1.10448   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 2.6022683 0.0066714 390.062 < 2e-16 \*\*\*  
## PC1 -0.1439179 0.0030985 -46.447 < 2e-16 \*\*\*  
## PC2 0.0229153 0.0042930 5.338 1.16e-07 \*\*\*  
## PC3 0.0005904 0.0045916 0.129 0.897705   
## PC4 0.0415341 0.0051992 7.989 3.66e-15 \*\*\*  
## PC5 -0.0181528 0.0058777 -3.088 0.002067 \*\*   
## PC6 0.1082711 0.0063294 17.106 < 2e-16 \*\*\*  
## PC7 0.0404840 0.0069386 5.835 7.23e-09 \*\*\*  
## PC8 -0.0122135 0.0070161 -1.741 0.082023 .   
## PC9 -0.0438688 0.0084337 -5.202 2.39e-07 \*\*\*  
## PC10 0.0649521 0.0091586 7.092 2.47e-12 \*\*\*  
## PC11 0.0220399 0.0099711 2.210 0.027300 \*   
## PC12 0.0988894 0.0110630 8.939 < 2e-16 \*\*\*  
## PC13 0.0425616 0.0122681 3.469 0.000544 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.2147 on 1022 degrees of freedom  
## Multiple R-squared: 0.73, Adjusted R-squared: 0.7266   
## F-statistic: 212.5 on 13 and 1022 DF, p-value: < 2.2e-16

set.seed(1000)  
# Run the PLS model  
pls\_school\_model = train(Average\_Proficiency ~ ., data = trainschool, method = "pls",   
 tuneLength = 10, preProcess = c("center", "scale"), trControl = trainControl(method = "repeatedcv",   
 number = 10))  
# summary(pls\_school\_model) head(pls\_school\_model$results)  
  
plot(pls\_school\_model, metric = "Rsquared")

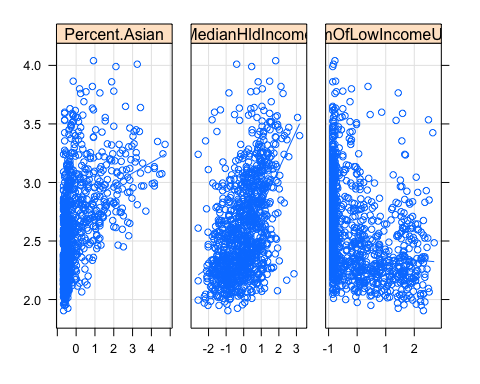
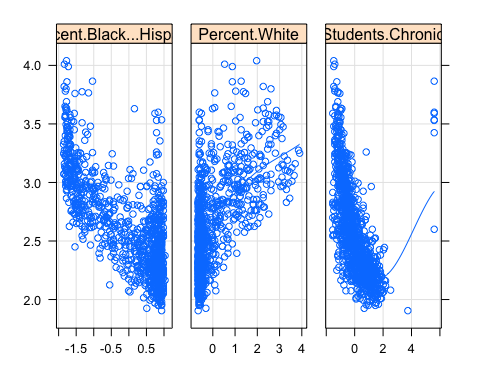


library(pls)  
pls\_school\_imp <- varImp(pls\_school\_model, scale = FALSE)  
plot(pls\_school\_imp, top = 10, scales = list(y = list(cex = 0.8)))



school\_order\_pls\_index <- order(abs(pls\_school\_imp$importance), decreasing = TRUE)  
top\_pls\_vars = rownames(pls\_school\_imp$importance)[school\_order\_pls\_index[c(1:6)]]  
top\_pls\_vars  
## [1] "Percent.Black...Hispanic"   
## [2] "Percent.White"   
## [3] "Percent.of.Students.Chronically.Absent"  
## [4] "Percent.Asian"   
## [5] "MedianHldIncome"   
## [6] "NumOfLowIncomeUnits"

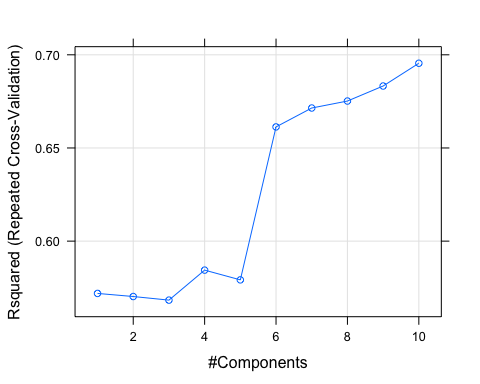
# Explore the univariate relationships of top 5 variables with Yield using  
# featureplot  
featurePlot(trainschool[, top\_pls\_vars], trainschool$Average\_Proficiency, plot = "scatter",   
 between = list(x = 1, y = 1), type = c("g", "p", "smooth"), layout = c(3,   
 1), labels = rep("", 2), warn = FALSE)



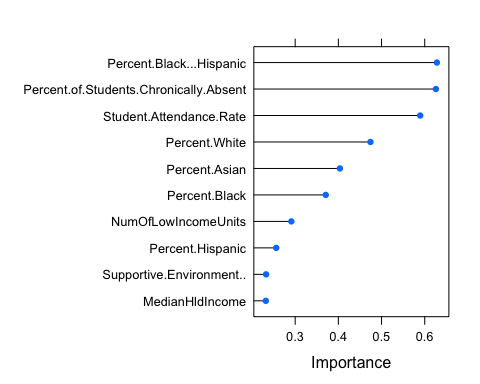
set.seed(1000)  
# Run the PCR model  
pcr\_school\_model = train(Average\_Proficiency ~ ., data = trainschool, method = "pcr",   
 tuneLength = 10, preProcess = c("center", "scale"), trControl = trainControl(method = "repeatedcv",   
 number = 10))  
summary(pcr\_school\_model)  
## Data: X dimension: 1036 18   
## Y dimension: 1036 1  
## Fit method: svdpc  
## Number of components considered: 10  
## TRAINING: % variance explained  
## 1 comps 2 comps 3 comps 4 comps 5 comps 6 comps 7 comps  
## X 25.78 39.21 50.95 60.10 67.27 73.45 78.59  
## .outcome 57.00 57.75 57.75 59.44 59.69 67.42 68.32  
## 8 comps 9 comps 10 comps  
## X 83.62 87.10 90.05  
## .outcome 68.40 69.11 70.44

head(pcr\_school\_model$results)  
## ncomp RMSE Rsquared MAE RMSESD RsquaredSD MAESD  
## 1 1 0.2699630 0.5719345 0.1936790 0.03295467 0.10659225 0.01676782  
## 2 2 0.2709151 0.5702755 0.1940351 0.02889337 0.10642116 0.01698404  
## 3 3 0.2717963 0.5683389 0.1941600 0.02950752 0.10826733 0.01715527  
## 4 4 0.2663478 0.5844361 0.1906140 0.03109414 0.10938448 0.01862537  
## 5 5 0.2679352 0.5792107 0.1944718 0.02866775 0.10326567 0.01964547  
## 6 6 0.2391812 0.6613000 0.1758450 0.02400737 0.06859397 0.01791496

plot(pcr\_school\_model, metric = "Rsquared")

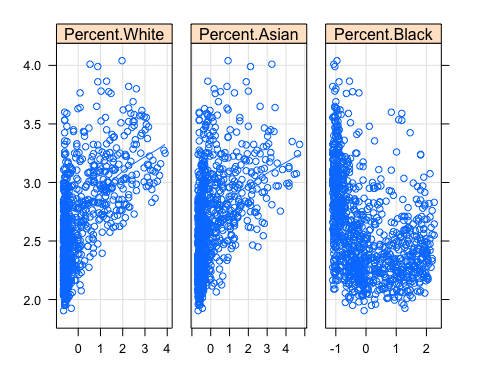
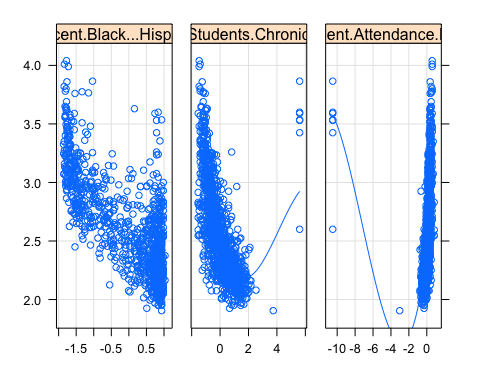


pcr\_school\_imp = varImp(pcr\_school\_model, scale = FALSE)  
plot(pcr\_school\_imp, top = 10, scales = list(y = list(cex = 0.8)))



school\_order\_pcr\_index <- order(abs(pcr\_school\_imp$importance), decreasing = TRUE)  
top\_pcr\_vars = rownames(pcr\_school\_imp$importance)[school\_order\_pcr\_index[c(1:6)]]  
top\_pcr\_vars  
## [1] "Percent.Black...Hispanic"   
## [2] "Percent.of.Students.Chronically.Absent"  
## [3] "Student.Attendance.Rate"   
## [4] "Percent.White"   
## [5] "Percent.Asian"   
## [6] "Percent.Black"

# Explore the univariate relationships of top 5 variables with Yield using  
# featureplot  
featurePlot(trainschool[, top\_pcr\_vars], trainschool$Average\_Proficiency, plot = "scatter",   
 between = list(x = 1, y = 1), type = c("g", "p", "smooth"), layout = c(3,   
 1), labels = rep("", 2), warn = FALSE)

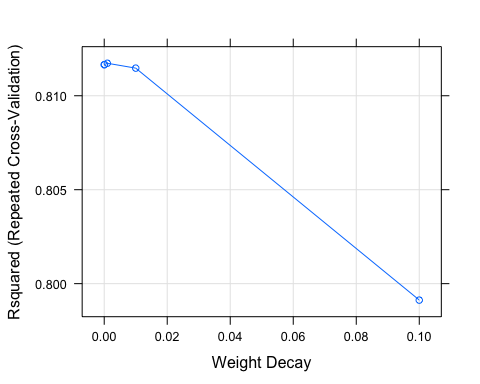


# Run the SVM model  
set.seed(1000)  
svm\_school\_model = train(Average\_Proficiency ~ ., data = trainschool, method = "svmLinear",   
 tuneLength = 10, trControl = trainControl(method = "repeatedcv", number = 10),   
 preProc = c("center", "scale"))  
# summary(svm\_school\_model)  
head(svm\_school\_model$results)  
## C RMSE Rsquared MAE RMSESD RsquaredSD MAESD  
## 1 1 0.1785456 0.8122949 0.1311427 0.01678941 0.03340694 0.01289993

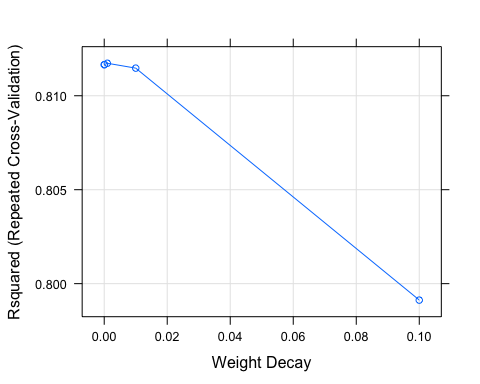
# Run the random forest model  
set.seed(1000)  
rf\_school\_model = train(Average\_Proficiency ~ ., data = trainschool, method = "rf",   
 tuneLength = 10, trControl = trainControl(method = "repeatedcv", number = 10),   
 preProc = c("center", "scale"))  
summary(rf\_school\_model)  
## Length Class Mode   
## call 4 -none- call   
## type 1 -none- character  
## predicted 1036 -none- numeric   
## mse 500 -none- numeric   
## rsq 500 -none- numeric   
## oob.times 1036 -none- numeric   
## importance 18 -none- numeric   
## importanceSD 0 -none- NULL   
## localImportance 0 -none- NULL   
## proximity 0 -none- NULL   
## ntree 1 -none- numeric   
## mtry 1 -none- numeric   
## forest 11 -none- list   
## coefs 0 -none- NULL   
## y 1036 -none- numeric   
## test 0 -none- NULL   
## inbag 0 -none- NULL   
## xNames 18 -none- character  
## problemType 1 -none- character  
## tuneValue 1 data.frame list   
## obsLevels 1 -none- logical   
## param 0 -none- list

head(rf\_school\_model$results)  
## mtry RMSE Rsquared MAE RMSESD RsquaredSD MAESD  
## 1 2 0.1824838 0.8106652 0.1363407 0.02274796 0.04187064 0.01258411  
## 2 3 0.1771680 0.8189351 0.1335407 0.02143973 0.03757408 0.01189464  
## 3 5 0.1747061 0.8221608 0.1325450 0.01957496 0.03384107 0.01124667  
## 4 7 0.1744152 0.8220463 0.1322082 0.01895921 0.03338747 0.01110054  
## 5 9 0.1741094 0.8222964 0.1316346 0.01801097 0.03153081 0.01066619  
## 6 10 0.1746740 0.8209536 0.1320336 0.01749881 0.03004029 0.01057376

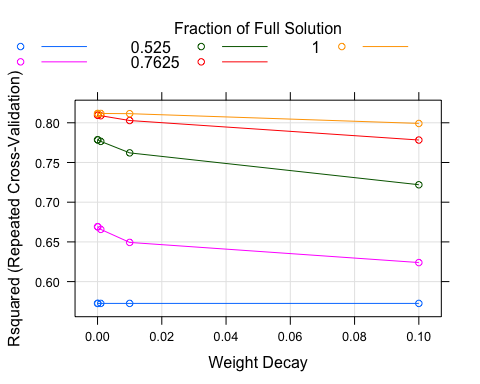
# Ridge Regression Model  
set.seed(1000)  
ridge\_school\_model <- train(Average\_Proficiency ~ ., data = trainschool, method = "ridge",   
 trControl = trainControl(method = "repeatedcv", repeats = 10), preProc = c("center",   
 "scale"), tuneLength = 5)  
# summary(ridge\_school\_model)  
  
plot(ridge\_school\_model, metric = "Rsquared")



# LASSO Model  
set.seed(1000)  
lasso\_school\_model <- train(Average\_Proficiency ~ ., data = trainschool, method = "lasso",   
 trControl = trainControl(method = "repeatedcv", repeats = 10), preProc = c("center",   
 "scale"), tuneLength = 5)  
# summary(lasso\_school\_model)  
plot(ridge\_school\_model, metric = "Rsquared")



# ENET Model (ELastic net Regression)  
set.seed(1000)  
library(elasticnet)  
enet\_school\_model <- train(Average\_Proficiency ~ ., data = trainschool, method = "enet",   
 trControl = trainControl(method = "repeatedcv", repeats = 10), preProc = c("center",   
 "scale"), tuneLength = 5)  
# summary(enet\_school\_model)  
plot(enet\_school\_model, metric = "Rsquared")



# Compare the model, this works only when trcontrol or sampling method is  
# same in all the models used.  
resamp\_school\_1 = resamples(list(lm = lm\_school\_model, lm.pca = lm\_pca\_school\_model,   
 pcr = pcr\_school\_model, rf = rf\_school\_model, svm = svm\_school\_model, pls = pls\_school\_model))  
print(summary(resamp\_school\_1))  
##   
## Call:  
## summary.resamples(object = resamp\_school\_1)  
##   
## Models: lm, lm.pca, pcr, rf, svm, pls   
## Number of resamples: 10   
##   
## MAE   
## Min. 1st Qu. Median Mean 3rd Qu. Max. NA's  
## lm 0.1089692 0.1233649 0.1359695 0.1321903 0.1399260 0.1501891 0  
## lm.pca 0.1379709 0.1489144 0.1668420 0.1633851 0.1754820 0.1905139 0  
## pcr 0.1489671 0.1576173 0.1697951 0.1710590 0.1814942 0.2043975 0  
## rf 0.1064348 0.1285058 0.1325998 0.1316346 0.1386385 0.1444307 0  
## svm 0.1101051 0.1214740 0.1365868 0.1311427 0.1395201 0.1492922 0  
## pls 0.1104177 0.1238209 0.1374505 0.1333441 0.1408000 0.1535184 0  
##   
## RMSE   
## Min. 1st Qu. Median Mean 3rd Qu. Max. NA's  
## lm 0.1589631 0.1645657 0.1757072 0.1785388 0.1852287 0.2107522 0  
## lm.pca 0.1932072 0.2024855 0.2169872 0.2189766 0.2330770 0.2560101 0  
## pcr 0.2047836 0.2122272 0.2254340 0.2270558 0.2320393 0.2741562 0  
## rf 0.1328847 0.1675912 0.1778096 0.1741094 0.1860899 0.1926468 0  
## svm 0.1596374 0.1649646 0.1765910 0.1785456 0.1849862 0.2100240 0  
## pls 0.1566127 0.1662412 0.1759248 0.1789738 0.1861559 0.2105733 0  
##   
## Rsquared   
## Min. 1st Qu. Median Mean 3rd Qu. Max. NA's  
## lm 0.7423889 0.8004919 0.8142882 0.8123980 0.8229791 0.8604814 0  
## lm.pca 0.6163403 0.7040741 0.7309941 0.7176772 0.7451179 0.7797917 0  
## pcr 0.5633670 0.6840057 0.7051075 0.6954602 0.7277260 0.7523079 0  
## rf 0.7696545 0.8039444 0.8259374 0.8222964 0.8434955 0.8750113 0  
## svm 0.7453879 0.7991412 0.8148354 0.8122949 0.8275070 0.8600858 0  
## pls 0.7428259 0.8015176 0.8110865 0.8116619 0.8223356 0.8627096 0

resamp\_school = resamples(list(ridge = ridge\_school\_model, lasso = lasso\_school\_model,   
 enet = enet\_school\_model))  
print(summary(resamp\_school))  
##   
## Call:  
## summary.resamples(object = resamp\_school)  
##   
## Models: ridge, lasso, enet   
## Number of resamples: 100   
##   
## MAE   
## Min. 1st Qu. Median Mean 3rd Qu. Max. NA's  
## ridge 0.1014926 0.1252876 0.1336853 0.1323358 0.1393254 0.1593956 0  
## lasso 0.1015560 0.1258310 0.1339568 0.1325472 0.1401016 0.1592786 0  
## enet 0.1014926 0.1252876 0.1336853 0.1323358 0.1393254 0.1593956 0  
##   
## RMSE   
## Min. 1st Qu. Median Mean 3rd Qu. Max. NA's  
## ridge 0.1424192 0.1660388 0.1764116 0.1785790 0.1912137 0.2273848 0  
## lasso 0.1417032 0.1663836 0.1770484 0.1786955 0.1915161 0.2269911 0  
## enet 0.1424192 0.1660388 0.1764116 0.1785790 0.1912137 0.2273848 0  
##   
## Rsquared   
## Min. 1st Qu. Median Mean 3rd Qu. Max. NA's  
## ridge 0.6742545 0.7825574 0.8143802 0.8117289 0.8444471 0.8878123 0  
## lasso 0.6750692 0.7816117 0.8141007 0.8114438 0.8441812 0.8860858 0  
## enet 0.6742545 0.7825574 0.8143802 0.8117289 0.8444471 0.8878123 0

print("RMSE an R2 for train data for these models:-")  
## [1] "RMSE an R2 for train data for these models:-"

Average\_Proficiency\_lm\_hat = predict(lm\_school\_model, newdata = subset(trainschool,   
 select = -c(Average\_Proficiency)))  
r2\_lm = cor(Average\_Proficiency\_lm\_hat, trainschool$Average\_Proficiency, method = "pearson")^2  
rmse\_lm = sqrt(mean((Average\_Proficiency\_lm\_hat - trainschool$Average\_Proficiency)^2))  
  
Average\_Proficiency\_pls\_hat = predict(pls\_school\_model, newdata = subset(trainschool,   
 select = -c(Average\_Proficiency)))  
r2\_pls = cor(Average\_Proficiency\_pls\_hat, trainschool$Average\_Proficiency, method = "pearson")^2  
rmse\_pls = sqrt(mean((Average\_Proficiency\_pls\_hat - trainschool$Average\_Proficiency)^2))  
  
Average\_Proficiency\_pcr\_hat = predict(pcr\_school\_model, newdata = subset(trainschool,   
 select = -c(Average\_Proficiency)))  
r2\_pcr = cor(Average\_Proficiency\_pcr\_hat, trainschool$Average\_Proficiency, method = "pearson")^2  
rmse\_pcr = sqrt(mean((Average\_Proficiency\_pcr\_hat - trainschool$Average\_Proficiency)^2))  
  
Average\_Proficiency\_rf\_hat = predict(rf\_school\_model, newdata = subset(trainschool,   
 select = -c(Average\_Proficiency)))  
r2\_rf = cor(Average\_Proficiency\_rf\_hat, trainschool$Average\_Proficiency, method = "pearson")^2  
rmse\_rf = sqrt(mean((Average\_Proficiency\_rf\_hat - trainschool$Average\_Proficiency)^2))  
  
Average\_Proficiency\_svm\_hat = predict(svm\_school\_model, newdata = subset(trainschool,   
 select = -c(Average\_Proficiency)))  
r2\_svm = cor(Average\_Proficiency\_svm\_hat, trainschool$Average\_Proficiency, method = "pearson")^2  
rmse\_svm = sqrt(mean((Average\_Proficiency\_svm\_hat - trainschool$Average\_Proficiency)^2))  
  
Average\_Proficiency\_ridge\_hat = predict(ridge\_school\_model, newdata = subset(trainschool,   
 select = -c(Average\_Proficiency)))  
r2\_ridge = cor(Average\_Proficiency\_ridge\_hat, trainschool$Average\_Proficiency,   
 method = "pearson")^2  
rmse\_ridge = sqrt(mean((Average\_Proficiency\_ridge\_hat - trainschool$Average\_Proficiency)^2))  
  
Average\_Proficiency\_lasso\_hat = predict(lasso\_school\_model, newdata = subset(trainschool,   
 select = -c(Average\_Proficiency)))  
r2\_lasso = cor(Average\_Proficiency\_lasso\_hat, trainschool$Average\_Proficiency,   
 method = "pearson")^2  
rmse\_lasso = sqrt(mean((Average\_Proficiency\_lasso\_hat - trainschool$Average\_Proficiency)^2))  
  
train.rmse.table <- rbind(rmse\_lm, rmse\_pls, rmse\_pcr, rmse\_rf, rmse\_svm, rmse\_ridge,   
 rmse\_lasso)  
train.rmse.table  
## [,1]  
## rmse\_lm 0.17456197  
## rmse\_pls 0.17530538  
## rmse\_pcr 0.22315110  
## rmse\_rf 0.07252326  
## rmse\_svm 0.17631498  
## rmse\_ridge 0.17456786  
## rmse\_lasso 0.17472930

train.r2.table <- rbind(r2\_lm, r2\_pls, r2\_pcr, r2\_rf, r2\_svm, r2\_ridge, r2\_lasso)  
train.r2.table  
## [,1]  
## r2\_lm 0.8191215  
## r2\_pls 0.8175776  
## r2\_pcr 0.7044127  
## r2\_rf 0.9718716  
## r2\_svm 0.8158980  
## r2\_ridge 0.8191093  
## r2\_lasso 0.8187865

print("RMSE an R2 for test data for these models:-")  
## [1] "RMSE an R2 for test data for these models:-"

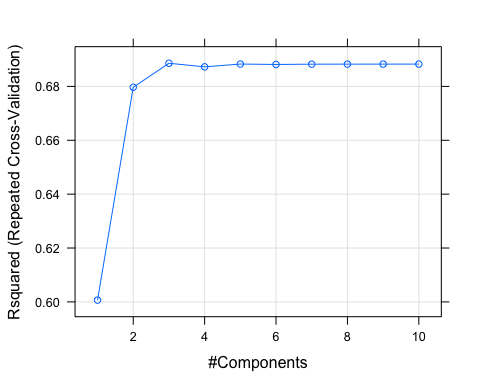
Average\_Proficiency\_lm\_hat = predict(lm\_school\_model, newdata = subset(testschool,   
 select = -c(Average\_Proficiency)))  
r2\_lm = cor(Average\_Proficiency\_lm\_hat, testschool$Average\_Proficiency, method = "pearson")^2  
rmse\_lm = sqrt(mean((Average\_Proficiency\_lm\_hat - testschool$Average\_Proficiency)^2))  
  
Average\_Proficiency\_pls\_hat = predict(pls\_school\_model, newdata = subset(testschool,   
 select = -c(Average\_Proficiency)))  
r2\_pls = cor(Average\_Proficiency\_pls\_hat, testschool$Average\_Proficiency, method = "pearson")^2  
rmse\_pls = sqrt(mean((Average\_Proficiency\_pls\_hat - testschool$Average\_Proficiency)^2))  
  
Average\_Proficiency\_pcr\_hat = predict(pcr\_school\_model, newdata = subset(testschool,   
 select = -c(Average\_Proficiency)))  
r2\_pcr = cor(Average\_Proficiency\_pcr\_hat, testschool$Average\_Proficiency, method = "pearson")^2  
rmse\_pcr = sqrt(mean((Average\_Proficiency\_pcr\_hat - testschool$Average\_Proficiency)^2))  
  
Average\_Proficiency\_rf\_hat = predict(rf\_school\_model, newdata = subset(testschool,   
 select = -c(Average\_Proficiency)))  
r2\_rf = cor(Average\_Proficiency\_rf\_hat, testschool$Average\_Proficiency, method = "pearson")^2  
rmse\_rf = sqrt(mean((Average\_Proficiency\_rf\_hat - testschool$Average\_Proficiency)^2))  
  
Average\_Proficiency\_svm\_hat = predict(svm\_school\_model, newdata = subset(testschool,   
 select = -c(Average\_Proficiency)))  
r2\_svm = cor(Average\_Proficiency\_svm\_hat, testschool$Average\_Proficiency, method = "pearson")^2  
rmse\_svm = sqrt(mean((Average\_Proficiency\_svm\_hat - testschool$Average\_Proficiency)^2))  
  
Average\_Proficiency\_ridge\_hat = predict(ridge\_school\_model, newdata = subset(testschool,   
 select = -c(Average\_Proficiency)))  
r2\_ridge = cor(Average\_Proficiency\_ridge\_hat, testschool$Average\_Proficiency,   
 method = "pearson")^2  
rmse\_ridge = sqrt(mean((Average\_Proficiency\_ridge\_hat - testschool$Average\_Proficiency)^2))  
  
Average\_Proficiency\_lasso\_hat = predict(lasso\_school\_model, newdata = subset(testschool,   
 select = -c(Average\_Proficiency)))  
r2\_lasso = cor(Average\_Proficiency\_lasso\_hat, testschool$Average\_Proficiency,   
 method = "pearson")^2  
rmse\_lasso = sqrt(mean((Average\_Proficiency\_lasso\_hat - testschool$Average\_Proficiency)^2))  
  
test.rmse.table <- rbind(rmse\_lm, rmse\_pls, rmse\_pcr, rmse\_rf, rmse\_svm, rmse\_ridge,   
 rmse\_lasso)  
test.rmse.table  
## [,1]  
## rmse\_lm 0.1871571  
## rmse\_pls 0.1876696  
## rmse\_pcr 0.2586807  
## rmse\_rf 0.1899804  
## rmse\_svm 0.1877061  
## rmse\_ridge 0.1871679  
## rmse\_lasso 0.1868665

test.r2.table <- rbind(r2\_lm, r2\_pls, r2\_pcr, r2\_rf, r2\_svm, r2\_ridge, r2\_lasso)  
test.r2.table  
## [,1]  
## r2\_lm 0.7909937  
## r2\_pls 0.7898114  
## r2\_pcr 0.6007476  
## r2\_rf 0.7855073  
## r2\_svm 0.7907787  
## r2\_ridge 0.7909586  
## r2\_lasso 0.7916075

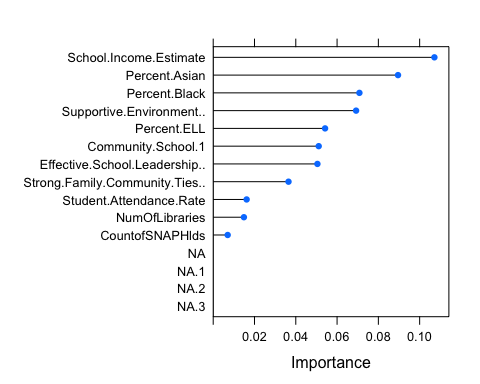
# Run Models on reduced 10 variables after rsq ratio based reduction dataset  
# Run the PLS model  
set.seed(10)  
pls\_school\_reduced\_model = train(Average\_Proficiency ~ ., data = trainreducedschool,   
 method = "pls", tuneLength = 10, preProcess = c("center", "scale"), trControl = trainControl(method = "repeatedcv",   
 number = 10))  
summary(pls\_school\_reduced\_model)  
## Data: X dimension: 1036 11   
## Y dimension: 1036 1  
## Fit method: oscorespls  
## Number of components considered: 3  
## TRAINING: % variance explained  
## 1 comps 2 comps 3 comps  
## X 21.05 30.31 40.01  
## .outcome 59.50 68.04 68.63

head(pls\_school\_reduced\_model$results)  
## ncomp RMSE Rsquared MAE RMSESD RsquaredSD MAESD  
## 1 1 0.2613701 0.6006576 0.1957567 0.05503631 0.11355916 0.03707651  
## 2 2 0.2337884 0.6796698 0.1789900 0.03920631 0.07473359 0.02716868  
## 3 3 0.2304139 0.6885915 0.1766444 0.03854458 0.07295851 0.02695374  
## 4 4 0.2310029 0.6872610 0.1768715 0.03968431 0.07523760 0.02663750  
## 5 5 0.2306637 0.6882995 0.1768463 0.03875560 0.07289035 0.02593737  
## 6 6 0.2307025 0.6881411 0.1766534 0.03877012 0.07302757 0.02584817

plot(pls\_school\_reduced\_model, metric = "Rsquared")

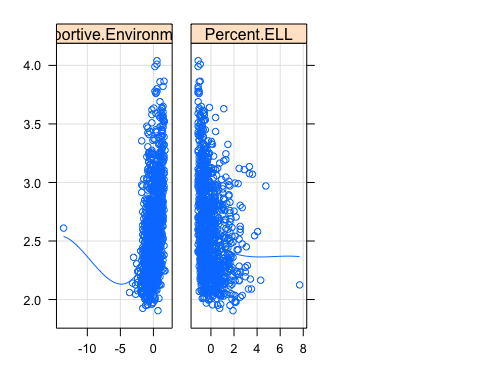
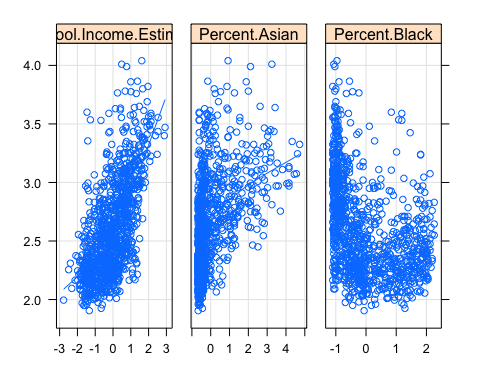


library(pls)  
pls\_school\_reduced\_imp <- varImp(pls\_school\_reduced\_model, scale = FALSE)  
plot(pls\_school\_reduced\_imp, top = 15, scales = list(y = list(cex = 0.8)))



reduced\_school\_order\_pls\_index <- order(abs(pls\_school\_reduced\_imp$importance),   
 decreasing = TRUE)  
reduced\_top\_pls\_vars = rownames(pls\_school\_reduced\_imp$importance)[reduced\_school\_order\_pls\_index[c(1:5)]]  
reduced\_top\_pls\_vars  
## [1] "School.Income.Estimate" "Percent.Asian"   
## [3] "Percent.Black" "Supportive.Environment.."  
## [5] "Percent.ELL"

# Explore the univariate relationships of top 5 variables with Yield using  
# featureplot  
featurePlot(trainreducedschool[, reduced\_top\_pls\_vars], trainreducedschool$Average\_Proficiency,   
 plot = "scatter", between = list(x = 1, y = 1), type = c("g", "p", "smooth"),   
 layout = c(3, 1), labels = rep("", 2), warn = FALSE)



Average\_Proficiency\_pls\_reduced\_hat = predict(pls\_school\_reduced\_model, newdata = subset(trainreducedschool,   
 select = -c(Average\_Proficiency)))  
r2\_train\_pls = cor(Average\_Proficiency\_pls\_reduced\_hat, trainreducedschool$Average\_Proficiency,   
 method = "pearson")^2  
r2\_train\_pls  
## [1] 0.6863462

rmse\_train\_pls = sqrt(mean((Average\_Proficiency\_pls\_reduced\_hat - trainreducedschool$Average\_Proficiency)^2))  
rmse\_train\_pls  
## [1] 0.2298695

Average\_Proficiency\_pls\_reduced\_hat = predict(pls\_school\_reduced\_model, newdata = subset(testreducedschool,   
 select = -c(Average\_Proficiency)))  
r2\_test\_pls = cor(Average\_Proficiency\_pls\_reduced\_hat, testreducedschool$Average\_Proficiency,   
 method = "pearson")^2  
r2\_test\_pls  
## [1] 0.6339481

rmse\_test\_pls = sqrt(mean((Average\_Proficiency\_pls\_reduced\_hat - testreducedschool$Average\_Proficiency)^2))  
rmse\_test\_pls  
## [1] 0.247569